Bridging the Divide: Reconsidering Softmax and Linear Attention

Dongchen Han* Yifan Pu* Zhuofan Xia* Yizeng Han Xuran Pan
Xiu Li Jiwen Lu Shiji Song Gao Huang

清華大學
Tsinghua University

NEURAL INFORMATION PROCESSING SYSTEMS

Code

Background

Transformers has a <u>Quadratic Complexity</u> $\mathcal{O}(N^2 d)$ *with respect to sequence length.*

High Resolution Images

Videos

Code

Background

Softmax Attention

- ✓ High expressive capability
- × **Quadratic complexity $\mathcal{O}(N^2 d)$**

$$S_i = \left[\frac{\exp\left(Q_i^\top K_1\right)}{\sum_{j=1}^{N}\exp\left(Q_i^\top K_j\right)}, \cdots, \frac{\exp\left(Q_i^\top K_N\right)}{\sum_{j=1}^{M}\exp\left(Q_i^\top K_j\right)}\right]^\top$$

$$O_i^S = S_i^\top V$$

Linear Attention

- × **Inferior performance**
- ✓ Linear complexity $\mathcal{O}(Nd^2)$

$$L_i = \left[\frac{\phi(Q_i)^\top \phi(K_1)}{\sum_{j=1}^{N}\phi(Q_i)^\top \phi(K_j)}, \cdots, \frac{\phi(Q_i)^\top \phi(K_N)}{\sum_{j=1}^{N}\phi(Q_i)^\top \phi(K_j)}\right]^\top$$

$$O_i^L = L_i^\top V = \frac{\phi(Q_i)^\top \left(\sum_{j=1}^{N}\phi(K_j) V_j^\top\right)}{\phi(Q_i)^\top \left(\sum_{j=1}^{N}\phi(K_j)\right)}$$

Injectivity of Attention Function

Code

Softmax Attn: $\mathrm{S_K}: \mathbb{R}^d \to \mathbb{R}^N, \quad \mathrm{S_K}(Q_i) = \left[\frac{\exp\left(Q_i^\top K_1\right)}{\sum_{j=1}^{N}\exp\left(Q_i^\top K_j\right)}, \cdots, \frac{\exp\left(Q_i^\top K_N\right)}{\sum_{j=1}^{M}\exp\left(Q_i^\top K_j\right)}\right]^\top$

Linear Attn: $\mathrm{L_K}: \mathbb{R}^d \to \mathbb{R}^N, \quad \mathrm{L_K}(Q_i) = \left[\frac{\phi(Q_i)^\top \phi(K_1)}{\sum_{j=1}^{N}\phi(Q_i)^\top \phi\left(K_j\right)}, \cdots, \frac{\phi(Q_i)^\top \phi(K_N)}{\sum_{j=1}^{N}\phi(Q_i)^\top \phi\left(K_j\right)}\right]^\top$

Proposition 1 -- Softmax Attn is Injective:

Given $K \in \mathbb{R}^{N\times d}$ *with* $\mathrm{rank(K)} = d, \mathrm{rank([K,1])} = d + 1. \forall p, q \in \mathbb{R}^d, p \neq q, we\ have\ \mathrm{S_K}(p) \neq \mathrm{S_K}(q).$

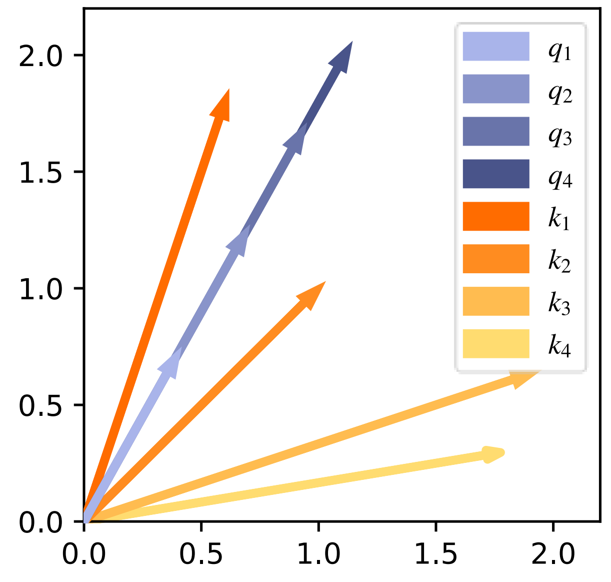
Proposition 2 -- Linear Attn is Not Injective:

Let $\varphi: \mathbb{R}^d \to \mathbb{R}^d$ *be acontinuous function.* $\exists p, q \in \mathbb{R}^d, p \neq q, \mathrm{s.t.}\ \mathrm{L_K}(p) = \mathrm{L_K}(q).$

Injectivity of Attention Function

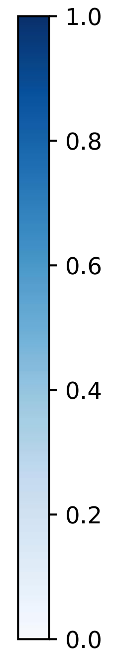
Code

$\phi(\cdot) = \mathrm{ReLU}(\cdot)$



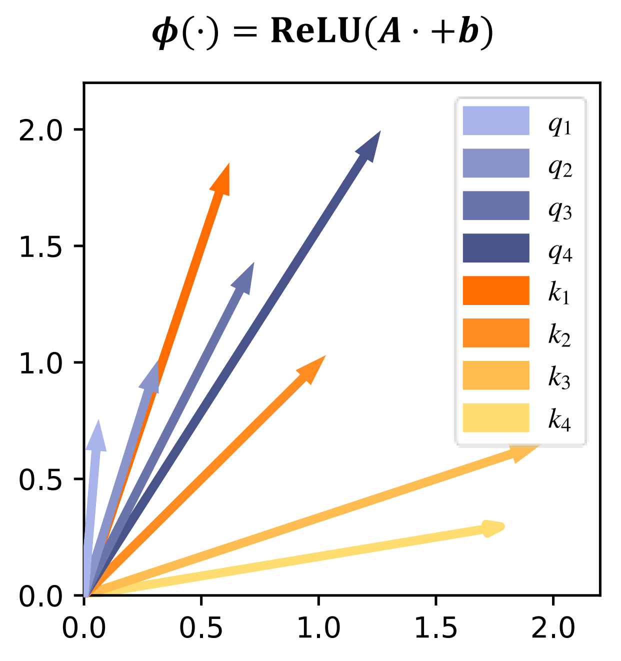
	$\mathrm{Attn}(q_1)$		$\mathrm{Attn}(q_2)$		$\mathrm{Attn}(q_3)$		$\mathrm{Attn}(q_4)$	
Softmax Attention	0.31	0.23	0.36	0.22	0.41	0.20	0.44	0.19
	0.25	0.20	0.25	0.17	0.24	0.15	0.23	0.13
Linear Attention	0.32	0.23	0.32	0.23	0.32	0.23	0.32	0.23
	0.25	0.19	0.25	0.19	0.25	0.19	0.25	0.19
InLine Attention (Ours)	0.31	0.23	0.35	0.22	0.39	0.21	0.42	0.21
	0.25	0.20	0.26	0.17	0.26	0.14	0.26	0.12

k_1	k_2
k_3	k_4



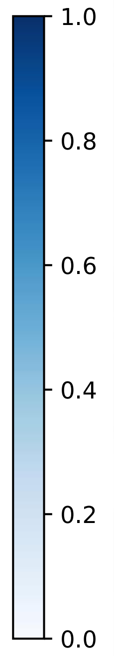
Injectivity of Attention Function

Code



	Attn(q_1)		Attn(q_2)		Attn(q_3)		Attn(q_4)	
Softmax Attention	0.36	0.25	0.37	0.23	0.39	0.21	0.41	0.19
	0.21	0.18	0.22	0.17	0.24	0.16	0.25	0.14
Linear Attention	0.31	0.24	0.31	0.24	0.31	0.24	0.31	0.24
	0.25	0.21	0.25	0.21	0.25	0.21	0.25	0.21
InLine Attention (Ours)	0.35	0.23	0.37	0.23	0.40	0.23	0.45	0.22
	0.24	0.18	0.24	0.16	0.24	0.13	0.24	0.10

k_1	k_2
k_3	k_4



Injective Linear Attention

Code

Injective Linear Attn:

$$\mathrm{InL_K}(Q_i) = [\phi(Q_i)^\top \phi(K_1), \cdots, \phi(Q_i)^\top \phi(K_N)]^\top - \frac{1}{N}\sum_{j=1}^{N} \phi(Q_i)^\top \phi(K_j) + \frac{1}{N}$$

Proposition 3 -- Injective Linear Attn is Injective:

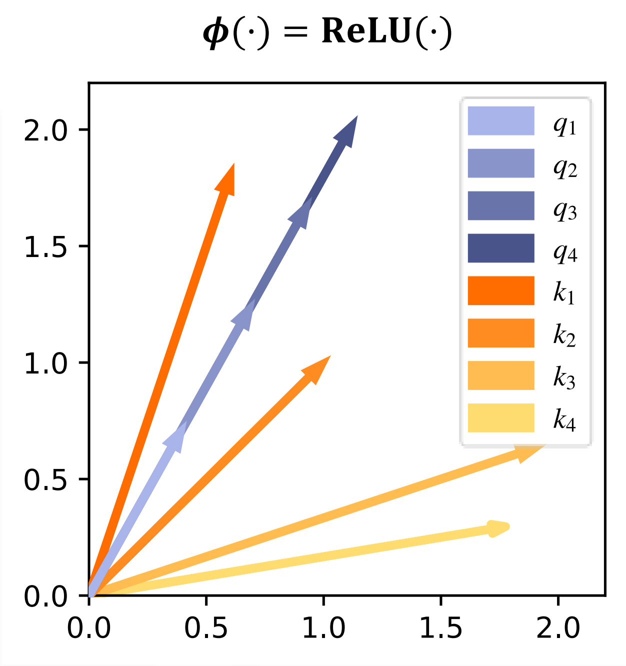
Let $\varphi: \mathbb{R}^d \to \mathbb{R}^d$ *be an injective map.*

Given $K \in \mathbb{R}^{N\times d}$ *with* $\mathrm{rank}(\phi(\mathrm{K})) = d, \mathrm{rank}([\phi(\mathrm{K}), 1]) = d + 1.$

$\forall p, q \in \mathbb{R}^d, p \neq q,$ *we have* $\mathrm{InL_K}(p) \neq \mathrm{InL_K}(q).$

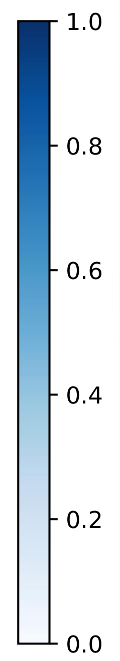
Injectivity of Attention Function

Code



	$\mathbf{Attn}(q_1)$		$\mathbf{Attn}(q_2)$		$\mathbf{Attn}(q_3)$		$\mathbf{Attn}(q_4)$	
Softmax Attention	0.31	0.23	0.36	0.22	0.41	0.20	0.44	0.19
	0.25	0.20	0.25	0.17	0.24	0.15	0.23	0.13
Linear Attention	0.32	0.23	0.32	0.23	0.32	0.23	0.32	0.23
	0.25	0.19	0.25	0.19	0.25	0.19	0.25	0.19
InLine Attention (Ours)	0.31	0.23	0.35	0.22	0.39	0.21	0.42	0.21
	0.25	0.20	0.26	0.17	0.26	0.14	0.26	0.12

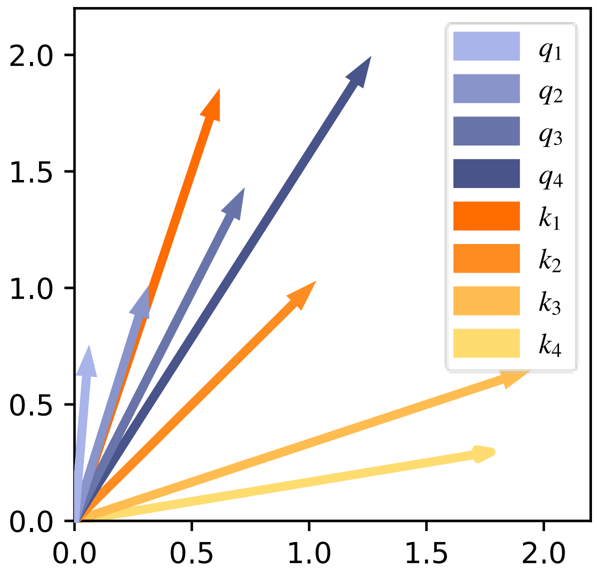
k_1	k_2
k_3	k_4



Injectivity of Attention Function

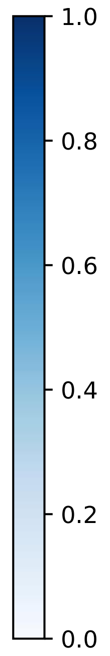
Code

$\boldsymbol{\phi}(\cdot) = \mathbf{ReLU}(\boldsymbol{A} \cdot + \boldsymbol{b})$



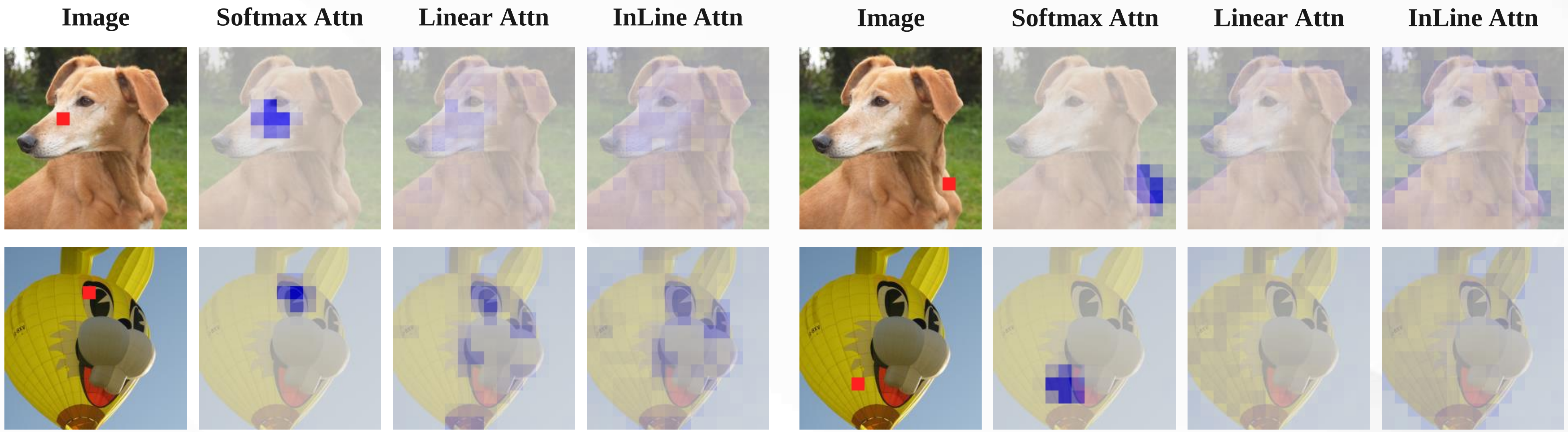
	Attn(q_1)		Attn(q_2)		Attn(q_3)		Attn(q_4)	
Softmax Attention	0.36	0.25	0.37	0.23	0.39	0.21	0.41	0.19
	0.21	0.18	0.22	0.17	0.24	0.16	0.25	0.14
Linear Attention	0.31	0.24	0.31	0.24	0.31	0.24	0.31	0.24
	0.25	0.21	0.25	0.21	0.25	0.21	0.25	0.21
InLine Attention (Ours)	0.35	0.23	0.37	0.23	0.40	0.23	0.45	0.22
	0.24	0.18	0.24	0.16	0.24	0.13	0.24	0.10

k_1	k_2
k_3	k_4



Code

Local Modeling Capability



Mask Out Position	None	Loc. 3×3	Loc. 5×5	Loc. 7×7	Rand 9	Rand 25	Rand 49
Softmax Attn	72.2	51.6	24.3	9.0	71.7	71.5	71.1
InLine Attn	70.0	58.0	40.0	20.0	70.0	69.9	69.5

Code

Local Modeling Capability

Injective Linear Attn:

$$\mathrm{InL}_{\mathrm{K}}(Q_i) = [\phi(Q_i)^\top\phi(K_1), \cdots, \phi(Q_i)^\top\phi(K_N)]^\top - \frac{1}{N}\sum_{j=1}^{N}\phi(Q_i)^\top\phi\left(K_j\right) + \frac{1}{N}$$

InLine Attention Module:

$$O_i = \mathrm{InL}_{\mathrm{K}}(Q_i)^\top V + \boxed{\sum_{j=1}^{9} r_j V_j^{N(i)}}, \quad r = \mathrm{MLP}(\bar{x})$$

Code

Empirical Study

✓ The impact of injective property:

Kernel Function $\phi(\cdot)$	$\mathrm{ReLU}(\cdot)$	$\mathrm{ReLU}(A\cdot+b)$	$\mathrm{LeakyReLU}(\cdot)$	$\mathrm{Identity}(\cdot)$
Linear Attn	77.3	70.2	1.5	0.2
InLine Attn	79.8	80.0	79.8	80.2

✓ Local modeling ability:

	Window	FLOPs	#Param	Acc.
InLine-Swin-T w/o res.	7^2	4.5G	30M	80.3
	14^2	4.5G	30M	80.4
	28^2	4.5G	30M	80.2
	56^2	4.5G	30M	80.2

	Window	FLOPs	#Param	Acc.
InLine-Swin-T w/ res.	7^2	4.5G	30M	81.6
	14^2	4.5G	30M	82.1
	28^2	4.5G	30M	82.3
	56^2	4.5G	30M	82.4

Code

Empirical Study

✓ Performances on ImageNet-1K:

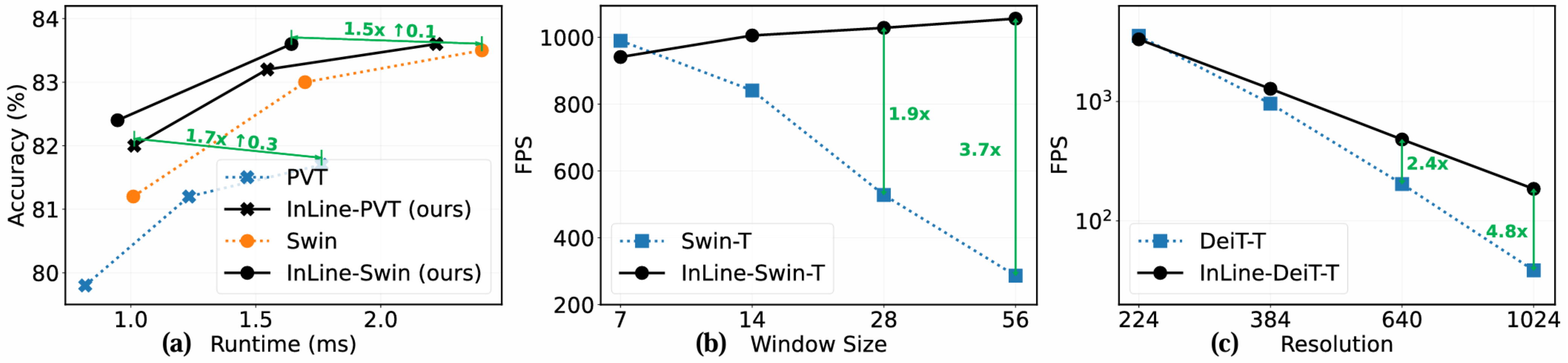
Method	Reso	#Params	FLOPs	Top-1
DeiT-T [30]	224^2	5.7M	1.2G	72.2
InLine-DeiT-T	224^2	6.5M	1.1G	**74.5** (+2.3)
DeiT-B	224^2	86.6M	17.6G	81.8
InLine-DeiT-B	448^2	23.8M	17.2G	**82.3** (+0.5)
PVT-S	224^2	24.5M	3.8G	79.8
InLine-PVT-S	224^2	21.6M	3.9G	**82.0** (+2.2)
PVT-L	224^2	61.4M	9.8G	81.7
InLine-PVT-L	224^2	50.2M	10.2G	**83.6** (+1.9)

Method	Reso	#Params	FLOPs	Top-1
Swin-T [19]	224^2	29M	4.5G	81.3
InLine-Swin-T	224^2	30M	4.5G	**82.4** (+1.1)
Swin-S	224^2	50M	8.7G	83.0
InLine-Swin-S	224^2	50M	8.7G	**83.6** (+0.6)
Swin-B	224^2	88M	15.4G	83.5
InLine-Swin-B	224^2	88M	15.4G	**84.1** (+0.6)
Swin-B	384^2	88M	47.0G	84.5
InLine-Swin-B	384^2	88M	45.2G	**85.0** (+0.5)

Empirical Study

Code

✓ Speed measurements:



Empirical Study

Code

✓ Performances on downstream tasks:

(a) Mask R-CNN Object Detection on COCO								
Method	FLOPs	Sch.	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
PVT-T	240G	1x	36.7	59.2	39.3	35.1	56.7	37.3
InLine-PVT-T	211G	1x	40.2	62.7	43.8	37.7	59.7	40.4
PVT-S	305G	1x	40.4	62.9	43.8	37.8	60.1	40.3
InLine-PVT-S	250G	1x	43.4	66.4	47.1	40.1	63.1	43.3
PVT-M	392G	1x	42.0	64.4	45.6	39.0	61.6	42.1
InLine-PVT-M	310G	1x	44.0	66.4	48.0	40.3	63.4	43.5
PVT-L	494G	1x	42.9	65.0	46.6	39.5	61.9	42.5
InLine-PVT-L	377G	1x	45.4	67.6	49.7	41.4	64.7	44.6
(b) Cascade Mask R-CNN Object Detection on COCO								
Method	FLOPs	Sch.	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
Swin-S	837G	3x	51.9	70.7	56.3	45.0	68.2	48.8
InLine-Swin-S	835G	3x	52.4	71.0	56.9	45.4	68.8	49.6
Swin-B	981G	3x	51.9	70.5	56.4	45.0	68.1	48.9
InLine-Swin-B	978G	3x	52.6	71.0	57.0	45.4	68.5	49.3

Code

Take-away Messages

Injective Linear Attention (InLine)

- ✓ The injectivity of attention function is of crucial importance
- ✓ Local modeling is essential to attention
- ✓ InLine: a simple, fast and effective linear attention module

Thank you!

Code

Contact: hdc23@mails.tsinghua.edu.cn